

REMARKS

Claims 1-49 were examined. Applicant has amended claims 1, 2, 14, 15, 38, 43 and 49. No claims are cancelled or are newly presented. No new matter has been introduced.

Amendments to the Drawings:

Figures 3 and 4 have been amended to include elements 32 and 56.

With regard to Figure 6, there is no element 54 in that Figure.

With regard to Figure 11, there is no element 381. There is an element 318.

Objections to the Specification

The specification is objected to because of formalities. Applicant has amended the specification. Paragraphs 43 and 46 has been amended.

Claim Objections

Claims 2 and 49 are objected to due to informalities. Applicant has amended the claims.

Rejections under 35 U.S.C. §101

Claims 1-49 are rejected under §101 as directed to non-statutory subject matter. Applicant has amended the claims.

Rejections under 35 U.S.C. §112

Claims 14 and 15 are rejected as lackign sufficient antecedent bases for the limitations of the claims. Applicant has amended the claims.

Rejections under 35 U.S.C. §103

Claims 1-10, 30-38, 42-43 and 47-49 stand rejected under §103(a) as being obvious over Rosaria (US 6,976,246).

Claims 11-29, 39-41 and 44-46 stand rejected under §103(a) as being obvious over Rosaria in view of Venter (2004/0194072).

This ground of rejection is respectively traversed.

In one embodiment of the present invention, as set forth in claim 1, a method is provided for generating test cases that are converted to an abstract representation. Semantic analysis is used to convert the test cases to abstract representations. A rule-based generation of test cases is provided from an abstract representation that includes application states, external interaction sequences and input data of test cases from data stores. Each application state is a set of application objects associated with a set of attributes and their values, or represents a runtime snapshot of an application under test which defines a context of external interaction. Generated test cases are validated. The test cases are converted to test scripts.

Rosaria discloses a system testing interface that provides a user with an interface to develop and generate a model for a software application under test, initiate the development of a test sequence with a graph traversal algorithm, and initiate the generation of a test execution program to test the application.

Unlike the present invention, Rosaria does not convert test cases into an internal abstract representation independent of the tools. Rosaria does not use semantic analysis of scripts based on an application model.

In Rosaria, a system testing interface is provided with a graphical user interface and a model generation engine to implement a model-based testing tool. The user interface enables a user to define a state table and the associated software application transitions in terms of simple user-defined rules. From the user-defined rules, the model generation engine generates the entire model (i.e., state table) of the software application under test. The state table is generated when the model generation engine calculates and develops all of the possible operational mode value combinations for the software application, and for each permutation, determines which software application inputs are applicable.

After the model of the software application under test is generated, the testing interface enables a user to select a graph traversal program that generates a test sequence of inputs for the software application from the model. The testing interface also enables a user to initiate a test of the application by selecting a test driver program

that executes the test sequence of inputs on the software application.

A rules editor enables a user to enter the parameters pertaining to transition information about the software application. The transition information includes a current state of the application which is associated with an input of the application, and a next state of the application. The next state of an application indicates the state of the application after the input has been applied to the current state of the application. A transition rule is defined by the current state, the input of the application, and the next state. A graph traversal menu enables a user to select from among multiple graph traversal algorithms and to generate a test sequence of application inputs to test the software application in a particular order of application inputs. A test execution menu enables a user to select from among multiple test driver programs and to initiate a test of the software application.

Venter provides for a multi-language compilation system and method. In Venter two or more separately written compiler components co-operate in the compilation of mixed language compilation units. Co-operation is facilitated through the use of the Common Compiler Infrastructure (CCI). A compiler for a specific language is written as extensions of base classes provided by the CCI. Common conventions and a flexible extensibility mechanism facilitate cooperation amongst compilers derived from the CCI.

Unlike the present invention, Venter does not convert test cases into an internal abstract representation independent of the tools. Venter does not use semantic analysis of scripts based on an application model.


In Venter, the base classes include visitor classes with visitor methods that are virtual and thus capable of being overridden. Visitor classes that derive from the base classes can add visitor methods to deal with new AST and IR node types that a particular compiler may need to introduce. When more than one language is present in a compilation unit more than one instance of a particular type of visitor may have to co-operate.

CONCLUSION

It is submitted that the present application is in form for allowance, and such action is respectfully requested. The Commissioner is authorized to charge any additional fees which may be required, including petition fees and extension of time fees, to Deposit Account No. 08-1641 (Docket No. 07464-0004).

Respectfully submitted,
HELLER EHRMAN LLP

Date: 4.3.07



Paul Davis, Reg. No. 29,294

275 Middlefield Road
Menlo Park, CA 94205
(650) 324-7041
Customer No. 25213